

Zentrale Grenzwertsatz

Zentrale Grenzwertsatz

Für jedes $n = 1, 2, \dots$ seien die Zufallsvariablen X_1, X_2, \dots, X_n unabhängig und besitzen die gleiche Verteilung mit dem Erwartungswert $\mu = E(X_i)$ und der Varianz $\sigma^2 = \text{Var}(X_i)$. Dann gilt für die Verteilungsfunktion der standardisierten Summen $G_n = \frac{\sum_{i=1}^n X_i - n\mu}{\sqrt{n}\sigma}$:

$$\lim_{n \rightarrow \infty} P(G_n \leq x) = \Phi(x)$$

Dabei ist Φ die Verteilungsfunktion der Standard-Normalverteilung $N(0, 1)$.

Überprüfen Sie mit R den zentralen Grenzwertsatz.

Verteilung

Erzeugen Sie einen Data Frame, der die gezogenen Zufallszahlen G_n enthält. Die Stichprobe soll einen Umfang von 50000 haben (Anzahl Zeilen der Tabelle). Der Data Frame hat sechs Spalten, die die Werte für die Zahlen $n = 1, 2, 3, 10, 100, 1000$ enthalten. Geben Sie die ersten 10 Zeilen des Data Frames aus.

Welche Verteilung Sie für X_i nehmen, bleibt Ihnen überlassen. Es soll nur keine Normalverteilung sein.

```
options(warn=-1)
# G_n: Umfang der Stichprobe
# n: Anzahl der Zufallszahlen

n_rows = 50000

g = data.frame(
  G_1 = rt(n_rows, 1),
  G_2 = rt(n_rows, 2),
  G_3 = rt(n_rows, 3),
  G_10 = rt(n_rows, 10),
  G_100 = rt(n_rows, 100),
  G_1000 = rt(n_rows, 1000)
)
print(dim(g))
```

```
## [1] 50000      6
```

```
print(head(g))
```

```
##           G_1           G_2           G_3           G_10           G_100           G_1000
## 1 -0.50651247 -0.92572105  1.3873983  0.4174636 -0.04092938  0.4944634
## 2 -2.56369497 -0.67087480  0.1376392  1.9257948  0.38167903  0.3079259
```

```
## 3  1.72606574  0.87644179 -0.3610023  0.1146069 -0.85105073  1.1895846
## 4 22.28851337  0.04072003  1.1351572 -1.8207777  0.50783374 -0.4173495
## 5  0.08645695 -0.42747464  1.4753047 -2.0641964  0.01608835  1.1691136
## 6 -0.03669335 -0.13711737 -0.9079204  0.7626160 -0.28306348 -2.0547245
```

Plot

Plotten Sie sechs Histogramme, die je für $n = 1, 2, 3, 10, 100, 1000$ die Verteilung im Vergleich zu einer $(0, 1)$ -Normalverteilung zeigen. Die Intervallbreite soll 0,25 sein.

Tipp: Platzieren Sie die sechs Plots auf einem 3×2 -Gitter.

```
library(ggplot2)
library(gridExtra)
library(grid)

normal_dist = data.frame(normal = rnorm(n_rows, 0, 1)) # Normalverteilung mit 50000 Zufallszahlen
alpha_t = 0.6

p1 = ggplot(g) +
  geom_histogram(data = normal_dist, aes(x = normal), binwidth = 0.25, color = "red", fill="transparent") +
  geom_histogram(aes(x = G_1), binwidth = 0.25, alpha = alpha_t) +
  xlim(-5, 5) +
  labs(title = "n = 1", x = "T-score", y = "count")

p2 = ggplot(g) +
  geom_histogram(data = normal_dist, aes(x = normal), binwidth = 0.25, color = "red", fill="transparent") +
  geom_histogram(aes(x = G_2), binwidth = 0.25, alpha = alpha_t) +
  xlim(-5, 5) +
  labs(title = "n = 2", x = "T-score", y = "count")

p3 = ggplot(g) +
  geom_histogram(data = normal_dist, aes(x = normal), binwidth = 0.25, color = "red", fill="transparent") +
  geom_histogram(aes(x = G_3), binwidth = 0.25, alpha = alpha_t) +
  xlim(-5, 5) +
  labs(title = "n = 3", x = "T-score", y = "count")

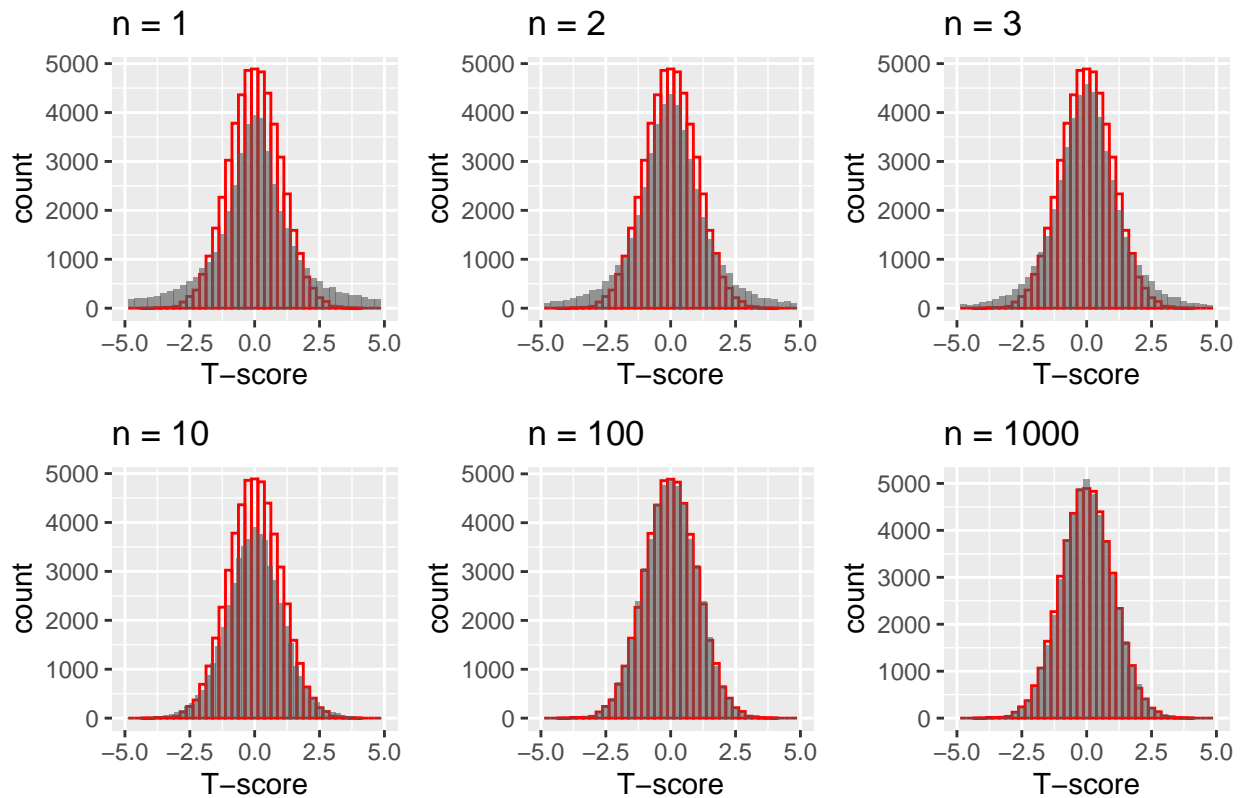
p4 = ggplot(g) +
  geom_histogram(data = normal_dist, aes(x = normal), binwidth = 0.25, color = "red", fill="transparent") +
  geom_histogram(aes(x = G_10), binwidth = 0.2, alpha = alpha_t) +
  xlim(-5, 5) +
  labs(title = "n = 10", x = "T-score", y = "count")

p5 = ggplot(g) +
  geom_histogram(data = normal_dist, aes(x = normal), binwidth = 0.25, color = "red", fill="transparent") +
  geom_histogram(aes(x = G_100), binwidth = 0.25, alpha = alpha_t) +
  xlim(-5, 5) +
  labs(title = "n = 100", x = "T-score", y = "count")

p6 = ggplot(g) +
  geom_histogram(data = normal_dist, aes(x = normal), binwidth = 0.25, color = "red", fill="transparent") +
  geom_histogram(aes(x = G_1000), binwidth = 0.25, alpha = alpha_t) +
  xlim(-5, 5) +
  labs(title = "n = 1000", x = "T-score", y = "count")
```

```
grid.arrange(p1, p2, p3, p4, p5, p6, ncol = 3, top=textGrob("Histogramme der Standardnormalverteilung (
```

Histogramme der Standardnormalverteilung (rot) und t-Verteilung (grau)



Abweichung

Nun soll die Abweichung der Standardisierten G von der Normalverteilung für alle n quantifiziert werden. Hierzu soll für alle Balken (bins) der Histogramme aus der vorigen Aufgabe die Differenz von G zur Normalverteilung gebildet und quadriert werden. Diese Werte werden aufaddiert und durch die Anzahl der Intervalle geteilt. Daraus wird die Wurzel gezogen.

Sie können sich auch ein anderes Maß zur Bestimmung der Abweichung überlegen.

Geben Sie die Abweichungen aus. Stimmt es, dass die Abweichungen mit größerem n kleiner werden?

Tipp: `hist(plot = FALSE)` erzeugt ein Histogramm, ohne es zu plotten. Gerne können Sie auch Ihr eigenes Histogramm nutzen.

```
library(knitr)

table = data.frame(n = c(1, 2, 3, 10, 100, 1000), mean_diff = c(0, 0, 0, 0, 0, 0))
for (i in 1:6) { # vergleich von jedem n mit Normalverteilung pro Bin
  histogram = hist(g[, i], plot = FALSE, binwidth = 0.25)
  histogram_normal = hist(normal_dist$normal, plot = FALSE, binwidth = 0.25)
  sqr_diff = (histogram$counts - histogram_normal$counts)^2
  mean_diff = sqrt(sum(sqr_diff) / length(sqr_diff))
  table[i, 2] = mean_diff
}
```

```
}  
print(kable(table))
```

```
##  
##  
## |      n| mean_diff|  
## |----:|-----:|  
## |     1| 8166.13890|  
## |     2| 8349.72817|  
## |     3| 8918.95733|  
## |    10| 3332.88975|  
## |   100|   64.10322|  
## |  1000| 1493.84398|
```

```
# Die Abweichungen werden kleiner, je größer n ist.
```